Spotify Music Suggestion Visualization

Zain Alam* Krishanu Datta† Jack Krolik‡ Lauren MacIver§ Neals Maniraja¶

ABSTRACT

Our visualization focuses on exploring new music through similar songs from Spotify's API. One of the domain tasks our visualization will support is the exploring and comparing of musical attributes. It is important to support this because it is useful for a user to be able to identify musical attributes of songs they like to find songs with similar musical attributes. The other domain task this visualization will support is song recommendation or the discovery of new songs. It is important to support this because it is important for users to be able to find new songs based on songs they currently like.

1 INTRODUCTION

Music preferences are subjective and with the magnitude of content on streaming platforms, finding new music can be a challenge. One of the domain tasks our visualization will support is the exploring and comparing of musical attributes. The user should be able to identify attributes of songs that they enjoy to determine what attributes draw them to a song and use those attributes to identify other songs they may enjoy. It is important to support this because it is useful for a user to be able to identify musical attributes of songs they like to find songs with similar musical attributes. Finding what musical attributes draw a user to a song (danceability, energy, etc.) is useful for helping users expand on their music taste. The other domain task this visualization will support is song recommendation or the discovery of new songs. Songs that have similar attributes are grouped together for the user to easily identify other songs that have the same desirable attributes. It is important to support this because it is useful for users to be able to find new songs based on songs they currently like. The end user for this visualization is anyone who is interested in learning more about their music preferences and wants to find more songs they would like to listen to. The data we are visualizing is the songs based on any two song features on the x and y axes in order to find how songs compare to each other based on certain attributes of said songs.

2 RELATED WORK

Grace [1] provides an overview of audio attributes in Spotify data and delves into many of the same features that we are interested in using for this project. This paper references to research about how the popularity of a song could be predicted based on various machine learning techniques and defines what different audio features are represented in the music. This paper focuses on interpreting audio feature data visually by drawing correlations between specific audio features, for example, loudness and energy have a strong positive correlation whereas loudness has a weak negative correlation with acoustic quality. In our visualization, we wish to give the user a holistic view of songs that are similar to each other based on selected attributes.

Additionally, Yu et al. [2] provide an in-depth analysis of matrix factorization for the purpose of determining optimal music recommendations. Something that this article touches on that we are not able to address with the current scope of our project is recommendations based on actual usage history and interaction data. Our visualization would require the user to have a prior understanding of their preferred music taste in order to accurately find similar songs, but we will consider how to make our visualization more available for our users to explore music because of this caveat. In our visualization, we wish to give the user the ability to select attributes from a dropdown menu to compare music.

3 USE CASE

Our visualization tool will allow target users to creatively analyze song data and dynamically discover new songs. We have two linked visual encodings in our visualization tool.

The first visual encoding, a scatterplot, will allow users to compare songs based on a selected pair of song attributes. The song attributes will constitute the axes of the scatterplot and the user will have the freedom to select the attributes represented by the axes via dropdown menus. As a result, users can locate a song they are interested in and utilize the visualization to find similar songs to it that they enjoy listening to based on certain features they prefer (i.e. loudness, danceability, tempo). Users are afforded additional information about each song simply by hovering over the point representative of the song on the scatterplot.

The second visual encoding, a bar graph, will allow users to further analyze information, specifically global market availability, about various songs. The bar graph will display the top markets available for the songs charted in the first visual encoding. Moreover, when users select songs in the first visualization the bars representative of the markets the selected song is available in will be highlighted in this encoding, revealing additional information about that song to the user. In brief, users are able to explore the geographic availability of various songs and observe trends in the availability of songs in certain markets.

One domain task that our visualization will support is exploring and comparing musical attributes. The user should be able to identify attributes of songs that they enjoy to filter other songs based on these attributes. This could be based on the genre, mood, or tempo of the song. The visualization should group together songs with similar attributes in order to do this.

Another domain task that our visualization will support is song recommendation or the discovery of new songs. Users should be able to visually identify what songs they are most likely to enjoy based on their current music taste. Songs that have similar attributes are grouped together for the user to easily identify other songs that have the same desirable attributes. The visualization should allow the user to adjust the attributes that they would like to use to identify their preferred new music recommendations.

These domain tasks will be accomplished with the visual encodings described above in section 3. Foremost, the scatterplot will be instrumental in aiding the user in both exploring musical attributes and discovering new songs. The fashion in which these domain tasks may be carried out is supported by the example usage scenario also found in section 3.

The intended user of this visualization tool is a music consumer seeking new songs for their listening pleasure. In providing the

*e-mail:alam.z@northeastern.edu
†e-mail:datta.k@northeastern.edu
‡e-mail:krolik.j@northeastern.edu
§e-mail:maciver.l@northeastern.edu
¶e-mail:maniraja.n@northeastern.edu

intended user with a scatterplot in which they can explore various songs based on chosen attributes, a user is able to select attributes they wish they build around in their music repertoire. Then, the user can utilize the resulting scatterplot to explore different groupings of songs based on similar attributes and add appealing songs to their music collection.

3.1 Example Usage Scenario

You are a 20-year-old college student who is tired of your repetitive and uninspiring music collection. You want to add more upbeat and lively songs that you have grown accustomed to listening to at parties. You discover our visualization tool which will perfectly suit your needs! Upon seeing our scatterplot and understanding the basics, you select danceability and tempo as your axes as you are searching for songs that are high in both those attributes. Looking in the top right of the graph, you find Pepas by Farruko which is exactly like the type of song you are searching for. Looking near Pepas on the graph, you find an array of songs with similar vibes. When you hover over them, the other visual encoding shows that they are available in your market, so you immediately add these songs to your collection and walk away with a more refreshed music palette.

4 DATA

The initial user of the dataset created a query to gather data from the Spotify API to help them identify what factors make a Spotify song popular. The query was run and data was collected from the Spotify API on 12/8/2020, using a query for every letter of the alphabet (Dataset Link). The original dataset contained 46 columns and 2000 rows. For the purposes of this assignment, a random subset of 2000 rows was selected using a Python script.

In terms of biases and ethical considerations, one facet is the columns that were collected. Since our study is reliant on a query written by another party, there might have been biasing in writing the query. Additionally, a consideration might be the bias present in Spotify's assignment of characteristics like valence which are more subjective and difficult to quantify. The popularity column might also have bias associated because it is pulled at a certain date which will influence results. Ethical considerations might include the manner in which Spotify is gathering/storing data from users and artists.

To clean the data, we first removed all columns that were not relevant to our analysis. For one, there was a lot of API data that were added as columns so these were deleted (i.e. album.external_urls.spotify, external_ids.isrc, external_urls.spotify). Additionally, columns that had data that was not unique, unuseful or repetitive for differentiating tracks were removed (i.e. time signature, track_number, disc_number). Next, we changed the 'key' column to include the actual key string (derived from the data dictionary in Spotify's API documentation) rather than the key id. Then, we replaced the album release date column data to just include the release year as some rows only included the year rather than the full date. Finally, we removed rows that contained missing values and renamed the column headers to be more readable

5 DESIGN PROCESS

Sketch #1 (Figure 1) depicts a scatter plot with points on the graph representing various songs in our dataset, and the circled song representing a song that the user has specifically selected. Along the x and y axes are two distinct quantitative song attributes. The songs that are in the same quadrant as the user-selected song, and consequently have the same color, are songs that are similar to the user-selected song by the attributes on the axes.

Sketch #1 (Figure 1) played a large part in influencing our final sketch (Figure 4). The visualization depicted in Figure 1 represents the base of one of our final visual encodings as the scatter plot with

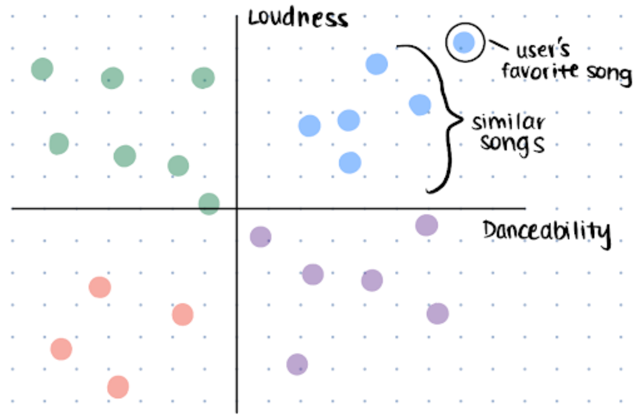

Figure 1: Rough Sketch 1

songs as points and attributes on the axes is a central piece of our final sketch (Figure 4).

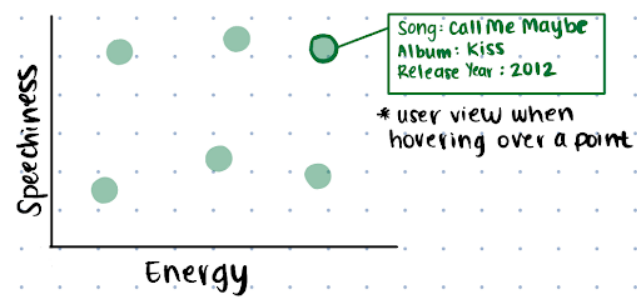

Figure 2: Rough Sketch 2

Sketch #2 (Figure 2) depicts a close-up of the action going on in Sketch #1 (Figure 1). In addition to the song attribute axes and colored song marks, in Sketch #2 (Figure 2), we decided to add hover interactivity to our visualization. Sketch #2 (Figure 2) shows that when a user hovers over a point (song) in the scatterplot, a tooltip will appear with additional information about the song, including but not limited to song name, song album, and song release year.

Sketch #2 (Figure 2) played a large role in influencing our final sketch (Figure 4) because we ultimately ended up including the tooltip functionality in our final visualizations. Sketch #2 (Figure 2) helped us realize we could provide the user with more information and flexibility in this implementation.

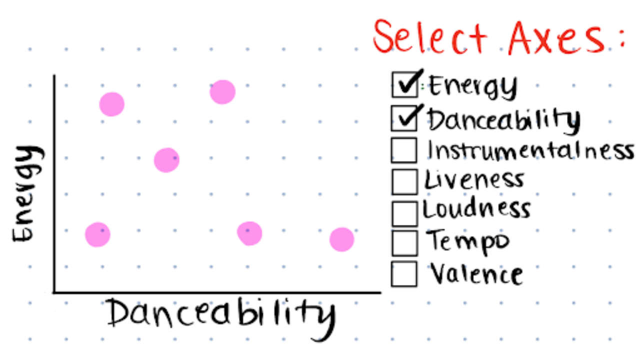

Figure 3: Rough Sketch 3

Sketch #3 (Figure 3) depicts an added component of the base visualization from Sketch #1 (Figure 1). Building on the scatter plot created in Sketch #1 (Figure 1), Sketch #3 (Figure 3) gives the graph a dynamic nature. It shows the possibility of swapping out the axis labels for whichever attributes the user wishes to visualize by utilizing a menu of checkboxes with many quantitative song attributes to choose from.

Sketch #3 (Figure 3) directly influenced our final sketch (Figure 4) as we decided to include the dynamic feature shown in Sketch #3 (Figure 3) in our final rendition. We opted to include the functionality in the form of dropdown menus to the right of our scatter plot where the user could select what attributes they want.

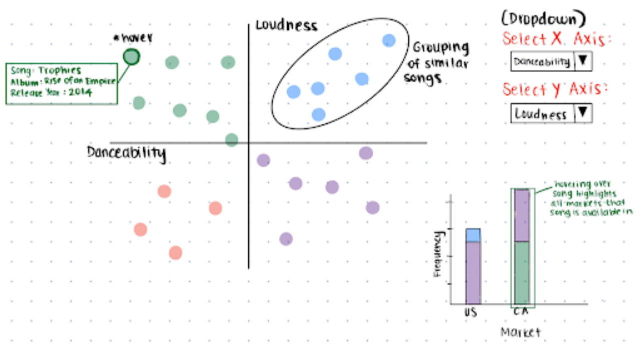

Figure 4: Final Sketch

Our final sketch (Figure 4) builds on and combines the work from all of our rough sketches (Figures 1, 2, and 3). The final sketch (Figure 4) includes two visual encodings.

The first encoding is a scatterplot with the marks being points and the channels being color and position. The points represent various songs, while their position on the graph conveys information about their attributes (which are labeled along the x and y axes). The color of the points conveys information about how similar the song is to the user-selected song based on the attributes selected for the axes. The user can gain more information about each song by hovering over the point on the graph. The song the user has specifically selected for analysis is highlighted with a circle. The user has the freedom to choose which attributes they want to be visualized with the dropdown menu for the axes on the right-hand side of the screen. In all the general functionality of this encoding is to allow users to find songs similar to their chosen songs dynamically and interactively.

The second visual encoding is a stacked bar graph, located in the bottom right corner of the final sketch (Figure 4), with the marks being the lines represented by the bars and the channels being vertical lengths, horizontal positions, and color. For each possible geographic market, a song in the first visualization is available in, there is a bar in the bar graph. The height of the bar represents how many songs in the visualization are available in that geographic market. The differing colors within each bar represent which area of the scatter plot visual encoding the song belongs to. Moreover, if all the songs in one bar are from the same quadrant, the bar would only have one color. An additional link between the two encodings occurs when a user hovers over a song in the scatterplot. When this happens, all the bars representing the geographic markets the song is available in, are highlighted. In all, the general functionality of this encoding is to show the user which markets the songs they are analyzing are available in and to help display the connection between the type of song (based on attributes) and geographic market.

6 Final Design

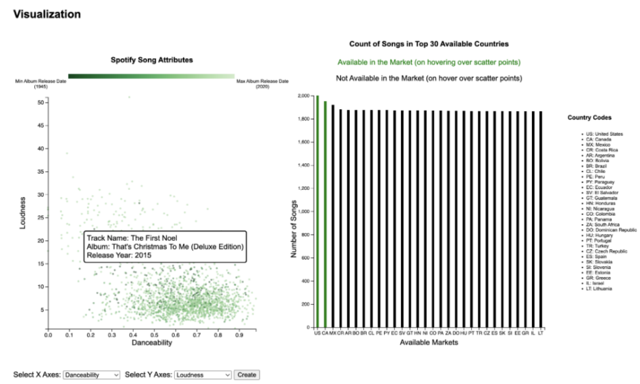

Figure 5: Final Visualization Image

Our final design (Figure 5) has a scatter plot showing the different attributes of Spotify songs. At the bottom of this graph, there are drop-down menus that allow you to select which attributes will be the axes for the scatter plot. In this graph, the marks are points through the channel of both horizontal and vertical positions. Our second visualization in our final design is a bar graph depicting the distribution of available markets of the songs shown in the scatter plot, where the length of the bar represents the number of songs available. Color as a channel is utilized in both of these visualizations. In the scatter plot, the gradient of the color of the point represents how long ago a song was released, and when hovering over one of the points, the bars for the countries where the song is available will be a different color. Additionally, both charts have tooltips for when the user hovers over a point or bar of the specifics of that mark.

One of the changes between the first sketches of our design and our final design includes changing the scatter plot graph to a two-axes grid as opposed to a four-quadrant grid. We found that since most of our attributes when plotted would be confined to a single quadrant, so to simplify the design to reflect this. In order to properly do this, we needed to convert our loudness attribute to positive values instead of negative ones so it would not skew our graph. Additionally, we decided to change our use of color. While we initially wanted points closer to each other to be grouped together, we decided to change what color should represent the release year of the song. This was to avoid redundantly encoding position and color to represent the song's similarities to other songs based on their attributes. Finally, in our bar graph depicting the markets, we experimented with using various numbers of markets and wanted to show the Top 50, but 30 markets looked the best visually for our usage.

After our usability testing, we were given feedback that there was no simple way to reference the total number of songs in each market, so a design change we implemented was adding a tool tip displaying this when the user hovers over a bar. Additionally, it was difficult to interpret the bar graph without knowing every country code, so we added a legend to the bar graph noting each code.

7 Discussion

Overall, our visualization was clear in showing the relationship of different Spotify features across 2,000 different songs. There were two main domain problems that we aimed to solve with our visualization tool, the first being a visualization that supports the exploring and comparing of musical attributes and the second being a visualization that recommends songs or helps users discover new songs. Our tool was successful in addressing both of these problems, as it visualized 2,000 different songs and allowed users to adjust which musical attributes the songs were graphed in terms. The visualization also supported a hover feature that allowed users to hover over nearby songs with similar features. Our visualization is somewhat limited by the data. There may be songs that are more similar to a given song, but because we only used 2,000 different songs it limits the pool of songs to choose from. Potential improvement considerations for the future might include expanding the number of songs used in our dataset to address the aforementioned limitation. Another possible addition could be a popup showing the 5 most similar songs based on the two selected features. This additional functionality would provide a more tangible recommendation system of new songs for users.

8 Conclusion

In this project, we designed and developed a visualization tool for analyzing and understanding the relationship between different variables in the given dataset. The tool allows users to explore the data, find patterns and correlations, and gain insights into the underlying trends. The tool is designed to support tasks such as data exploration, hypothesis generation, and trend analysis. It is important to support these tasks as they help users make informed decisions based on data and improve their understanding of the domain. The target user for this tool is anyone who needs to analyze data and gain insights, such as data analysts, researchers, and business professionals. The data used for this tool is a public dataset collected by Kaggle, and we performed data cleaning and derived attributes to make it suitable for visualization. The final visualization tool consists of multiple views coordinated to provide a comprehensive analysis of the data, with the final design evolved during the implementation process and informed by usability testing. The tool fully addresses the domain problem of understanding the relationship between variables in the dataset, but there are limitations in terms of the complexity of the data that can be analyzed. In the future, improvements to the tool could include more advanced analytics and interactivity, and integration with other tools to support decision-making. Overall, this project contributes to the field of data visualization by providing a useful tool for analyzing and understanding songs' data. For main contributions, Lauren MacIver and Nealaksi Maniraja created the bar chart, Krishanu Datta created the tooltip, Zain Alam helped create the scatterplot and functionality to allow axes to be interchangeable, and Jack Krolik created the scatterplot and did all the data web scraping and cleaning.

9 Appendix

9.1 Data Abstraction

Each item (row) in our dataset represents a song.

Below is a table of the attribute types of the attributes on our dataset.

Attribute Name	Attribute Type
Danceability	Ordered, Quantitative, Sequential
duration	Ordered, Quantitative, Sequential
energy	Ordered, Quantitative, Sequential
instrumentalness	Ordered, Quantitative, Sequential
liveness	Ordered, Quantitative, Sequential
loudness	Ordered, Quantitative, Sequential
speechiness	Ordered, Quantitative, Sequential
tempo	Ordered, Quantitative, Sequential
valence	Ordered, Quantitative, Sequential
popularity	Ordered, Quantitative, Sequential
album.total tracks	Ordered, Quantitative, Sequential
album.release date	Ordered, Ordinal, Sequential
key	Ordered, Ordinal, Cyclic
mode	Categorical
album.name	Categorical
album.available markets	Categorical
album.album type	Categorical
name	Categorical
available markets	Categorical
explicit	Categorical

9.2 Task Abstraction

Music platforms like Spotify have a plethora of songs, which sometimes makes it difficult to navigate and find songs that you may enjoy. Thus, we hope that our visualization will help solve that problem by using musical attributes of songs liked by the user to find similar songs based on the attributes.

In order to compare musical attributes so the user can filter other songs based on the attributes of songs they enjoy. The visualization should group together songs that have their user-selected attributes most similar to the songs they enjoy.

In order to recommend songs for the user, the user needs to identify songs they currently like as well as their preferred attributes of the song, and the visualization should use those songs' attributes to recommend similar songs to the user by grouping those songs with the songs they like based on those attributes. The visualization should also allow the user to adjust the attributes and have the groupings change accordingly.

Our visualization tool has three levels of tasks that cater to the different needs and interests of its users. At the high level, the tool is used for consumption, allowing users to discover music that they may be interested in. The mid-level task is focused on searching, where users can explore similarities or differences in their preferred music and see the available markets it's in. At the low level, users can identify a single target, such as a song that has similar attributes to a different song they enjoy. Additionally, they can summarize the song's availability in different markets using the tool's bar chart.

The tool's targets are the aspects of data that are of interest to the user. In this case, the features of the songs are important to the users because they help classify songs differently. Overall, the tool provides a comprehensive solution for music enthusiasts who are interested in discovering new music, exploring similarities or differences in their preferred music, and understanding the availability of music in different markets.

References

[1] A. S. Grace. Song and artist attributes analysis for spotify. *2022 International Conference on Engineering and Emerging Technologies (ICEET)*, 2022. doi: 10.1109/ICEET56468.2022.10007360

[2] Y. Yu, S. Tang, A. M. Jorge, and A. H. Nabizadeh. Predicting user preference based on matrix factorization by exploiting music attributes. *Proceedings of the Ninth International C* Conference on Computer Science Software Engineering - C3S2E '16*, 2016. doi: 10.1145/2948992.2949010